

PC Centronic 24 line programmable I/O card (CIO-1)

User's Manual

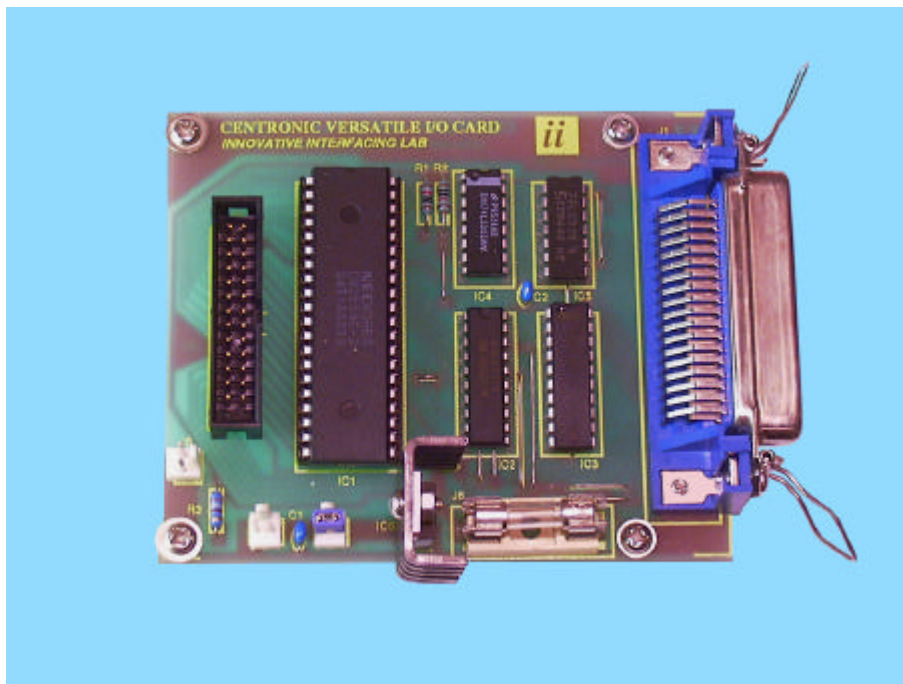
The CIO-1 Centronic 24-line programmable I/O card is a general-purpose input/output card. It is connected to a computer via its Centronic port. As almost all desktop PCs and notebook PCs have at least one Centronic port on-board, this card is a universal solution for adding I/O interfaces to PCs.

The card offers 24 input/output lines. These lines are organised into 4 groups: Port A, Port B, Upper four bits of Port C (Port CU) and Lower four bits of Port C (Port CL). Each group can be configured as an input or output port. The 24 I/O lines are available from a 26-way male DIL PCB header for easy connection.

The card requires a 7-15V DC power supply. A 1A fuse is provided on the card. The card consumes approximately 20 mA on its own.

A DOS-based program "CIODos" and 16-bit and 32-bit Windows 95/98 program "CIOWin" are supplied with the I/O card. The programs allow users to check functionality of the card.

A Turbo Pascal 6 unit, 16-bit and 32-bit Windows Dynamic Link Libraries (DLLs) are supplied with the I/O card. They contain basic I/O procedures. After declaring them in user's programs, users can call them anywhere in their TP6 or Windows' 16-bit or 32-bit programs.



Centronic 24-line programmable I/O card, CIO-1

1. Hardware

1.1 I/O lines

The card has 24 input/output lines. These lines are organised into 4 groups: Port A, Port B, Upper four bits of Port C (Port CU) and Lower four bits of Port C (Port CL). Each port can be configured as an input or output. Each line in a port can not be configured individually. I/O lines are TTL compatible.

NOTE: The maximum voltage input to the card is 5V. This card is not designed for connecting live voltages.

Port group	Number of lines	Description
Port A	8	Can be configured as an 8-bit input or output port
Port B	8	Can be configured as an 8-bit input or output port
Port CU	4	Can be configured as an 4-bit input or output port
Port CL	4	Can be configured as an 4-bit input or output port

All I/O lines are available from a 26-way DIL male header (see Figure 1 for pin function details).

1.2 Data transfer rate

Data write and read rate depends on the host computer. For a Pentium 233MHz PC, a data transfer rate of 70kHz can be achieved.

1.3 Power supply

The card requires a 7-15V DC (1A) voltage supply. An on-board 1A fuse protects the over load of current of the power supply. The card provides connectors for power input, a power switch and an LED indicator. The CIO-1 card consumes 20 mA current. The regulated +5V power supply is available from the 26-way DIL socket.

1.4 Sizes and basic connections

The size of the card is 10 cm by 8.5 cm. Figure 1 shows the basic connections of the card.

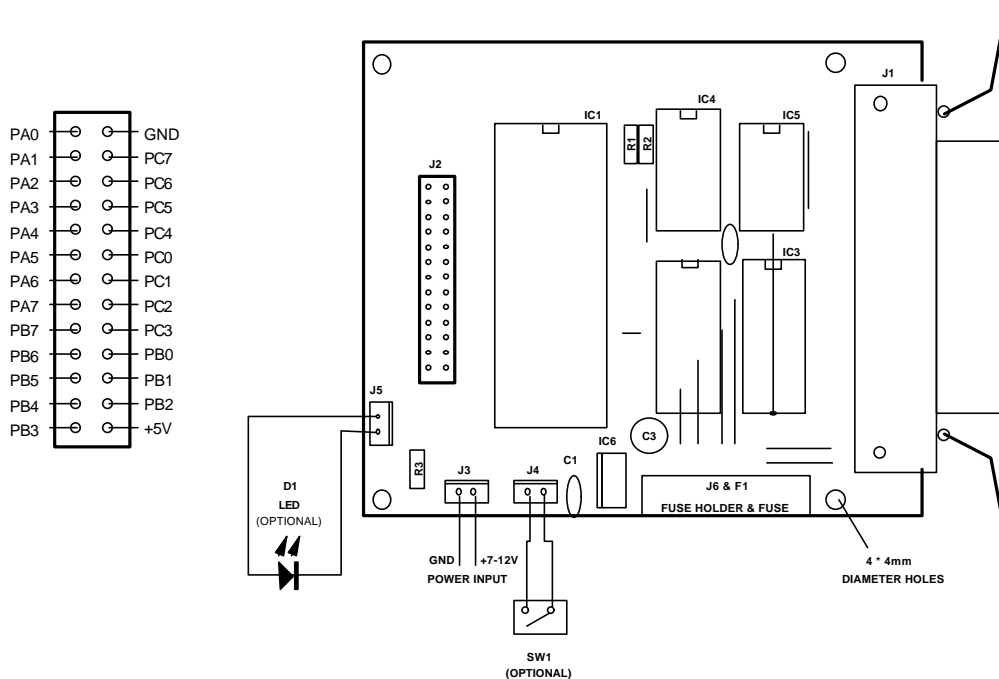


Figure 1 CIO-1 card and details of the connectors

2. Application note

The Centronic is one of the most common I/O ports that every modern computer has.

The *Centronic port*, also known as the *printer port* or the *parallel port*, is an industrial standard interface designed for connecting printers to a computer. A computer at least has one such a port installed. The port may come with computer's mother-boards or come with plug-in I/O cards. Adding more Centronic ports is easy and inexpensive. In total, four Centronic ports may be installed on a computer and they have logic names LPT1 to LPT4.

Originally, the port was designed for connecting computers to printers. They can also be used for other interfacing applications. Peripheral devices designed for the Centronic port not only provide the easiest way of connection to computers but also offer a universal hardware solution for all computers.

The CIO-1 card allows the Centronic port to be used. It serves as an interface between the computer and external circuits. The card utilises an industrial standard peripheral programmable interface IC (PPI) 8255. The IC has 24 input/output lines. The lines are organised into 4 groups: Port A, Port B, Upper four bits of Port C (Port CU) and Lower four bits of Port C (Port CL). Each port can be configured as an input or output port. To configure the port, a byte should be written into the control register of the 8255. The following table summarises the possible control word.

Control word	Port A	Port C (Upper)	Port B	Port C (Lower)
128	Output	Output	Output	Output
129	Output	Output	Output	Input
136	Output	Input	Output	Output
137	Output	Input	Output	Input
130	Output	Output	Input	Output
131	Output	Output	Input	Input
138	Output	Input	Input	Output
139	Output	Input	Input	Input
144	Input	Output	Output	Output
145	Input	Output	Output	Input
152	Input	Input	Output	Output
153	Input	Input	Output	Input
146	Input	Output	Input	Output
147	Input	Output	Input	Input
154	Input	Input	Input	Output
155	Input	Input	Input	Input

After configuration, you can write data to an output port or read data from an input port. The I/O drivers provided with the I/O card allow users to configure ports, write data and read data from ports.

Using these drivers, users do not need to know any hardware details of the card. They only need to decide which ports are to be configured as inputs or outputs by writing an appropriate control word to the CIO card. Then they can write data into output ports and read data from input ports. See Section 4 for details of I/O drivers.

3. Software

The floppy disk supplied with the card contains the following directories:

- \Manual
- \Drivers
- \Dos
- \Win16
- \Win32
- \Labview

Files in each directory are listed as follows:

Files	Description	Explanation
\Manual Directory		
Manual.doc	Manual in MSWord 97 format	
CIOTP6.pas	Turbo Pascal 6 source codes for I/O functions. This program produces CIOTP6.tpu	For users which want to convert the TP6 program into other programming languages such as Basic or C++
\Drivers Directory		
CIOTP6.tpu	A TP6 unit containing I/O functions	They contain all I/O functions to control the card.
CIO16.dll	16-bit Windows DLLs	
\Dos Directory		
CIODos	Dos-based user program	Test the functionality of the I/O card in DOS environment
CIOTP6.tpu	A TP6 unit containing I/O functions	
\Win16, Win32 and Labview Directories		
This directory contains necessary files for installation. Following the installation wizard.		Test the functionality of the I/O card in Windows environment

4. TP6 and Windows DLL functions

There are 4 TP6 I/O, 4 16-bit DLL and 3 32-bit DLL functions to control all the operations of the CIO card. The same function in the TP6 unit and 16-bit Windows DLLs shares an identical name. All functions are listed below. For 32-bit DLLs, the first function "Centronic(x)" is not available.

Function Name	Description
<i>Centronic(x)</i> <i>(NOTE: Not available for 32-bit DLLs)</i>	X=0: return number of LPT installed on a PC X=1: return base address of 1 st LPT port (in decimal) X=2: return base address of 2 nd LPT port (in decimal) X=3: return base address of 3 rd LPT port (in decimal)
<i>Write_control(P_addr, Byte)</i>	Output control word (<i>Byte</i>) to the Control register to configure ports
<i>Write_port(P_addr, Port_number, Byte)</i>	Output a byte to an output port. The port is specified by <i>Port_number</i> (0,1 or 2). You can only write data to a port that is configured as output.
<i>Inputbyte(P_addr, Port_number)</i>	Input a byte from an input port. The port is specified by <i>Port_number</i> (0,1 or 2). You can only read data from a port that is configured as input.

Note:

1. *P_addr* (declared as a word) is the base address of the selected LPT port (in decimal). *P_addr* is obtained using the following command: *P_addr = Centronic(x)* (for TP6 and 16-bit Windows DLLs, x=1 or 2 or 3 for LPT1, LPT2 or LPT3).
2. *Port_number* (declared as a byte) is 0, 1 or 2
3. *Byte* (declared as a byte) is an 8-bit wide byte

The TP6 program for I/O functions are given in Program List 1. How the functions are used in user's program is demonstrated in a TP6 and a VB4 program sample.

4.1 TP6 I/O functions

TP6 program "CioTP6.Pas" contains all I/O functions to control the CIO card. The complete program source code is listed as follows.

Program List 1

```
(*****
*
* Module:      Intec Centronic 8255 I/O card - CIO-1 card
*
* Author:      Pei An
*              Intec Associates Limited
*              11 Sandpiper Dirve
*              Stockport
*              SK3 8UL UK
*              Tel. +44-(0)161 477 9583
*              Fax. +44-(0)161 477 5755
*              Email: pan@intec-group.co.uk
*
* Copyright 1999 Intec Associates Limited
*
* Description:
*
* This module provides functions to drive the Intec Centronic 8255 I/O card
*
* Centronic connection details:
*
* Functions included:
* -Centronic(x): select a Centronic port and find the address
* -Out1(P_address,data): Output data from Data port of the selected Centornic port
```

```

-Output(P_address,data); Output data from Control port of the selected Centronic port
-Function convert(bytex:integer):integer;
-Function write_control(P_address, Control_word:integer): integer;
-Function Write_port (P_address,Port_number,Output_byte:integer): integer;
-Function Inputbyte(P_address, Port_number:integer):integer;

***** *)
{$DEFINE MAIN}
{$UNDEF MAIN}

{$IFDEF MAIN}
program CIOTP6;

{$ELSE}
unit CIOTP6;

Interface
    Function Centronic(x:integer):integer;
    Function Out1(P_address,data:integer):integer;
    Function Out2(P_address,data:integer):integer;
    Function convert(bytex:integer):integer;
    Function write_control(P_address, Control_word:integer): integer;
    Function Write_port (P_address,Port_number,Output_byte:integer): integer;
    Function Inputbyte(P_address, Port_number:integer):integer;

Implementation
{$ENDIF}

Function Centronic(x:integer):integer;
(* find the number of Centronic port installed and select a base address
$000:$0408 holds the printer base address for LPT1
$000:$040A holds the printer base address for LPT2
$000:$040C holds the printer base address for LPT3
$000:$040E holds the printer base address for LPT4
$000:$0411 number of parallel interfaces in binary format *)
var
    number_of_LPT, LPT1, LPT2, LPT3, LPT4 :integer;
begin
    number_of_LPT:=mem[$40:$11];          (* read number of parallel ports *)
    number_of_LPT:=(number_of_lpt and (128+64)) shr 6;
    lpt1:=0; lpt2:=0; lpt3:=0; lpt4:=0;
    LPT1:=memw[$40:$08];                 (* Memory read procedure *)
    LPT2:=memw[$40:$0A];
    LPT3:=memw[$40:$0C];
    LPT4:=memw[$40:$0E];
    case x of
        0:   centronic:=number_of_LPT;
        1:   centronic:=lpt1;
        2:   centronic:=lpt2;
        3:   centronic:=lpt3;
        4:   centronic:=lpt4
    end;
end;

Function Out1(P_address,data:integer):integer;
(* output from the data port of centronic port *)
var
    dummy:integer;
begin
    port[P_address]:=data;
end;

Function Out2(P_address,data:integer):integer;
(* output from the control port of centronic port *)
var
    dummy:integer;
begin
    port[P_address+2]:=data;
    for dummy:=1 to 80 do dummy:=dummy;
end;

Function convert(bytex:integer):integer;
(* CONVERT THE bytex INTO THE BYTE TO BE SENT INTO THE 8255 REGISTERS
byte received by the 8255:          (byte into 8255 = B8 B7 B6 B5 B4 B3 B2 B1)
byte output from the Centronic port: (data from Centronic port = B8 B4 B7 B3 B6 B2 B5 B1)
The difference in bit sequence is due to hardware connection of the board. To sent a byte
into the 8255, data from the Centronic port should take the above form *)

```

```

begin
  convert:= ((bytex and 1) + ((bytex and 2) shl 1)+
            ((bytex and 4) shl 2) + (bytex and 8) shl 3+
            ((bytex and 16) shr 3) + ((bytex and 32) shr 2) +
            (bytex and 64) shr 1 + ((bytex and 128)));
end;

Function write_control(P_address, Control_word:integer): integer;
(* Control_word is output by port [P_address], 8255 R/W is controlled by port [P_address+2]
Port [P_address+2] bit configuration, -1(-Read) -2(-Write) 3(A0) -4(A1) *)
var
  dummy:integer;
begin
  Control_word:=convert(Control_word);(*Convert the word*)
  dummy:=Out1(P_address,Control_word); (* Control word output from port [P_address]*)
  dummy:=Out2(P_address,0+0+4+0);      (*-Read=1, -Write=1, A0=1, A1=1, delay=10 ms*)
  dummy:=Out2(P_address,0+2+4+0);      (*-Read=1, -Write=0, A0=1, A1=1, delay=10 ms*)
  dummy:=Out2(P_address,0+0+4+0);      (*-Read=1, -Write=1, A0=1, A1=1, delay=10 ms*)
end;

Function Write_port (P_address,Port_number,Output_byte:integer): integer;
(* WRITE A BYTE (output_byte) TO A 8255 IC PORT SPECIFIED BY port_number
Output byte is output by port [P_address], 8255 R/W is controlled by port [P_address+2]
Port [P_address+2] bit configuration, -1(-Read) -2(-Write) 3(A0) -4(A1) *)
var
  dummy:integer;
begin
  Output_byte:=convert(Output_byte); (*Convert the word*)
  dummy:=Out1(P_address,Output_byte); (*Output a byte from port [P_address]*)
  if (Port_number=0) then begin (*Output a byte to A register of 8255 IC*)
    dummy:=Out2(P_address,0+0+0+8); (*-Read=1, -Write=1, A0=1, A1=0*)
    dummy:=Out2(P_address,0+2+0+8); (*-Read=1, -Write=0, A0=1, A1=0*)
    dummy:=Out2(P_address,0+0+0+8); (*-Read=1, -Write=1, A0=1, A1=0*)
  end;
  if (Port_number=1) then begin (*Output a byte to B register of 8255 IC*)
    dummy:=Out2(P_address,0+0+4+8);
    dummy:=Out2(P_address,0+2+4+8);
    dummy:=Out2(P_address,0+0+4+8);
  end;
  if (Port_number=2) then begin (*Output a byte to C register of 8255 IC*)
    dummy:=Out2(P_address,0+0+0+0);
    dummy:=Out2(P_address,0+2+0+0);
    dummy:=Out2(P_address,0+0+0+0);
  end;
end;

Function Inputbyte(P_address, Port_number:integer):integer;
(* READ A BYTE (input_byte) FROM A 8255 IC PORT SPECIFIED BY port_number
Input byte is input from port [P_address+1], 8-bit byte is input into the Centronic
port in two stages. In the first stage, bits 1,2,3 and 4 are read and in the
second one bits 5,6,7 and 8 are read. This operation is controlled by bit 1
of port [P_address]. 8255 R/W is controlled by port [P_address+2]
Port [P_address+2] bit configuration, -1(-Read) -2(-Write) 3(A0) -4(A1) *)
var
  byte_1st, byte_2nd:byte;
  dummy:integer;
begin
  if (Port_number=0) then begin (*Read a byte from port A of 8255*)
    dummy:=Out1(P_address,0); (*DSL=0, Prepare to read low 4 bits*)
    dummy:=Out2(P_address,0+0+0+8); (*-Read=1, -Write=1, A0=0, A1=0*)
    dummy:=Out2(P_address,1+0+0+8); (*-Read=0, -Write=1, A0=0, A1=0*)
    byte_2nd:= port[P_address+1]; (*Read low 4 bits*)
    dummy:=Out1(P_address,1); (*DSL=1, Prepare to read high 4 bits*)
    byte_1st:=port[P_address+1]; (*Read high 4 bits*)
    dummy:=Out2(P_address,0+0+0+8); (*-Read=1, -Write=1, A0=0, A1=0*)
  end;

  if (Port_number=1) then begin (*Read a byte from port B of 8255*)
    dummy:=Out1(P_address,0);
    dummy:=Out2(P_address,0+0+4+8);
    dummy:=Out2(P_address,1+0+4+8);
    byte_2nd:=port[P_address+1];
    dummy:=Out1(P_address,1);
    byte_1st:=port[P_address+1];
    dummy:=Out2(P_address,0+0+4+8);
  end;

```

```

if (Port_number=2) then begin          (*Read a byte from port C of 8255*)
  dummy:=Out1(P_address,0);
  dummy:=Out2(P_address,0+0+0+0);
  dummy:=Out2(P_address,1+0+0+0);
  byte_2nd:=port[P_address+1];
  dummy:=Out1(P_address,1);
  byte_1st:=port[P_address+1];
  dummy:=Out2(P_address,0+0+0+0);
  end;
  (*Note:  Byte_1st  =(x B8 B7 B6 B5 x x x)
           Byte_2nd  =(x B4 B3 B2 B1 x x x)
           x=don't care, B8 to B1: bit value 0 or 1*)

  byte_1st:=byte_1st and 120; (*x B8 B7 B6 B5 x x x and 01111000 = 0 B8 B7 B6 B5 0 0 0 0*)
  byte_1st:=byte_1st shl 1;  (*shift 1 bit left, byte_1st =B8 B7 B6 B5 0 0 0 0*)
  byte_2nd:=byte_2nd and 120; (*x B4 B3 B2 B1 x x x and 01111000 = 0 B4 B3 B2 B1 0 0 0 0*)
  byte_2nd:=byte_2nd shr 3;  (*shift 3 bit right, byte_2nd =0 0 0 0 B4 B3 B2 B1*)
  Inputbyte:=byte_1st or byte_2nd; (*byte_1st or byte_2nd = B8 B7 B6 B5 B4 B3 B2 B1*)
end;
begin
end.

```

4.2 How a Turbo Pascal 6 program calls TP6 I/O functions

To write a TP6 user program, CioTP6.TPU unit has to be declared at the beginning of the program. Then I/O functions can be called elsewhere within the program. A sample program is shown in Program list 2. This program gives similar user interface as the CioDos program.

Program List 2

```

Program Centronic_8255_interface;
(* Demonstration program for Cen8255 I/O card
   This program shows how to write data to or read data from the Cen8255 I/O card
   PA = Input, PB and PC = Outputs. Control Word = 144

   Some useful control words: 128(decimal): Port A, B and C all as output
                               155(decimal): Port A, B and C all as input
                               144(decimal): Port A as input; Port B and C as output
                               146(decimal): Port A and B as input; Port C as output *)

uses
  crt,dos,CIOTP6;
{ CIOTP6 contains the I/O procedures controlling the card }

var
  command,output_byte,input_byte,bitnumber,portnumber,outputbyte:byte;
  P_address,delaynumber,dummy:integer;

Procedure find_delay_number;
(* Check pc speed and find the delaynumber for 1 ms *)
var
  time1,time2,dt:real;
  t,h1,m1,s1,s1001,h2,m2,s2,s1002:word;
begin
  clrscr;
  gotoxy(30,24); write('Checking computer speed');
  gettime(h1,m1,s1,s1001);
  time1:=3600*h1+60*m1+s1+s1001/100;
  for t:=1 to 50000 do delay(1);
  gettime(h2,m2,s2,s1002);
  time2:=3600*h2+60*m2+s2+s1002/100;
  dt:=time2-time1;
  delaynumber:=round(50000/dt*0.001);
  clrscr;
  gotoxy(30,24); write('Finished...');
  clrscr;
end;

Procedure Select_printer_address;
(* select a printer port *)
var
  Number_LPTs,i,Selected_LPT:integer;
begin
  writeln;
  writeln;
  if Centronic(Centronic(0))=0 then Number_LPTs:=Centronic(0)-1 else Number_LPTs:=Centronic(0);

```

```

if Centronic(Centronic(0)-1)=0 then Number_LPTs:=Centronic(0)-2 else Number_LPTs:=number_LPTs;
textcolor(Lightred);
writeln('
                                Centronic Port Report of your PC');
writeln;
writeln;
textcolor(yellow);
write('
                                ', Number_LPTs, ' Centronic port(s) installed on your PC:');
writeln;
writeln;
textcolor(green);
for i:=1 to Number_LPTs do writeln('
                                Base address of LPT',i,' =
',Centronic(i),' Decimal ');
textbackground(Green);
textcolor(magenta);
repeat
gotoxy(24,20); write('Input 1,2 or 3 to select an LPT : '); readln(Selected_LPT);
until (Selected_LPT>=0) and (selected_LPT<4);
P_address:=Centronic(Selected_LPT);
end;

```

```

Procedure find_bit_weight;

```

```

(* Find the bit weight *)

```

```

begin

```

```

if bitnumber=1 then outputbyte:=1; {find the bit weight of the selected bit}
if bitnumber=2 then outputbyte:=2;
if bitnumber=3 then outputbyte:=4;
if bitnumber=4 then outputbyte:=8;
if bitnumber=5 then outputbyte:=16;
if bitnumber=6 then outputbyte:=32;
if bitnumber=7 then outputbyte:=64;
if bitnumber=8 then outputbyte:=128;

```

```

end;

```

```

{*****MAIN PROGRAM*****}

```

```

begin

```

```

find_delay_number;
Select_printer_address;
textbackground(blue);
textcolor(yellow);
clrscr;

```

```

repeat

```

```

clrscr;
textbackground(cyan);
write ('
                                Centronic port CIO-1 card testing program

```

```

');

```

```

textbackground(blue);
writeln;
writeln;
writeln('
                                Ports B and C are OUTPUT');
writeln('
                                Port A is INPUT');
writeln;
writeln('
                                The control word sent to the 8255 PPI is 144 decimal');
dummy:=write_control(P_address,144);
writeln;
textcolor(lightmagenta);
write ('
                                Input the bit number to be tested (1 to 8, 9 to quit) : ');

```

```

readln(Bitnumber);

```

```

writeln;
textcolor(lightRed);
gotoxy(13,23); write('The selected bit of Ports B and C produces a square wave');
gotoxy(13,24); write(' Value input from Port A is shown on the screen');
gotoxy(30,25); write('Press Return to continue');
textcolor(lightcyan);
Find_bit_weight;
repeat

```

```

dummy:=write_port(P_address,1, outputbyte);
dummy:=write_port(P_address,2, outputbyte);
Gotoxy(30,20); write('Data from Port A =',Inputbyte(P_address,0):3);
delay(delaynumber*300);

```

```

dummy:=write_port(P_address,1,0);
dummy:=write_port(P_address,2,0);
Gotoxy(30,20); write('Data from Port A =',Inputbyte(P_address,0):3);
delay(delaynumber*300);

```

```

until keypressed or (bitnumber=9);

```

```

until bitnumber=9;

```

```

end.

```

4.3 How a Visual Basic program calls 16-bit DLL functions

This Visual Basic 4 (16-bit system) program consists of a form as shown in Figure 6 (CioWin). The screen dump of the form is shown in Figure 6. The complete source codes are listed in Program list 3.

All 4 I/O DLL functions should be declared first. Then they can be called in other part of the program. The DLLs are 16-bit. They can be only used for 16-bit programming environment.

Program list 3

```
'Declare functions in the dynamic link library
'Declared functions: Centronic(), Inputbyte(), Write_port() and write_control()
Private Declare Function Centronic Lib "c:\windows\system\w8255dll.dll" (ByVal X As Integer) As Integer
Private Declare Function Write_control Lib "c:\windows\system\w8255dll.dll" (ByVal P_address As Integer, ByVal
Output_data As Integer) As Integer
Private Declare Function Write_port Lib "c:\windows\system\w8255dll.dll" (ByVal P_address As Integer, ByVal
address As Integer, ByVal Output_data As Integer) As Integer
Private Declare Function Inputbyte Lib "c:\windows\system\w8255dll.dll" (ByVal P_address As Integer, ByVal address
As Integer) As Integer

Dim status(30) As Integer
Dim P_address As Integer

Function bit_weight(bitnumber As Integer)
If bitnumber = 1 Then bit_weight = 1
If bitnumber = 2 Then bit_weight = 2
If bitnumber = 3 Then bit_weight = 4
If bitnumber = 4 Then bit_weight = 8
If bitnumber = 5 Then bit_weight = 16
If bitnumber = 6 Then bit_weight = 32
If bitnumber = 7 Then bit_weight = 64
If bitnumber = 8 Then bit_weight = 128
End Function

Private Sub Command1_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
Label3.Caption = " Get the status of the Status port inputs (4 inputs)"
End Sub

Private Sub Command2_Click(Index As Integer)
'Change status of the outputs of Data port and Control port
status(Index) = 1 - status(Index) 'toggle action
If status(Index) = 1 Then Shape1(Index).BackColor = &HFF& Else Shape1(Index).BackColor = black 'output status
to the virtual LEDs
End Sub

Private Sub Command2_MouseMove(Index As Integer, Button As Integer, Shift As Integer, X As Single, Y As Single)
'show on-line help when mouse pointer movers across the buttons
If Index <= 7 Then
Label3.Caption = " Change status of port A outputs (8 outputs)"
End If
If Index > 7 And Index < 15 Then
Label3.Caption = " Change status of port B outputs (8 outputs)"
End If
End Sub

Private Sub Command3_Click()
'Re-select a Centronic port
dummy = MsgBox(Str(Centronic(0)) - 1 & " Centronic ports (LPTs) are installed on your PC. Their base
addresses are: " & Format$(Centronic(1), "###") & " " & Format$(Centronic(2), "###") & " " &
Format$(Centronic(3), "###") & " " & Format$(Centronic(4), "###") & "Decimal", 48, "Centronic ports (LPT) on
your PC") 'show information on installed LPTs
lpt_number = Val(InputBox$("Input 1, 2, 3 or 4 to select a Centronic port (LPT) for the Mini-Lab Data Logger/
Controller", "Select LPT ports")) 'Select a Centronic port
P_address = Centronic(lpt_number) 'find the base address of the selected LPT port
dummy = Write_control(P_address, 137) 'Configure the 8255, port A and B as outputs, port C as input

Label2(0).Caption = "Selected LPT port : " & Format(lpt_number) 'show the information on the selected
LPT port
Label4(0).Caption = "Base address of LPT: " & Format(P_address) 'show the information of the selected LPT
port
End Sub

Private Sub Command3_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
'show on-line help when mouse pointer moves across the button
Label3.Caption = "Re-select Centronic port"
End Sub
```

```

Private Sub Command4_Click()
    'Quit the program
    End
End Sub

Private Sub Command4_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    Label3.Caption = " Quit the program"
End Sub

Private Sub Command5_Click()
    'Output data to the Data port
    Output_byte = 0
    For i = 0 To 7
        Output_byte = Output_byte + status(i) * bit_weight(i + 1) 'form a byte to be output
    Next i
    dummy = Write_port(P_address, 0, Output_byte) 'Output the byte to the Data port

    'Output data to the Data port
    Output_byte = 0
    For i = 8 To 15
        Output_byte = Output_byte + status(i) * bit_weight(i - 7) 'form a byte to be output
    Next i
    dummy = Write_port(P_address, 1, Output_byte) 'output the byte to the Control port
    input_byte = Inputbyte(P_address, 2)
    For i = 16 To 23
        status(i) = (input_byte And bit_weight(i - 15)) / bit_weight(i - 15)
        If status(i) = 1 Then Shape1(i).BackColor = &HFF& Else Shape1(i).BackColor = black
    Next i
End Sub

Private Sub Command5_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    Label3.Caption = " Output data to Port A and Port B and read data from Port C"
End Sub

Private Sub Form_Load()
    'initialize status()
    'show Centronic port information and allow user to select an LPT
    dummy = MsgBox(Str(Centronic(0) - 1) & " Centronic ports (LPTs) are installed on your PC. Their base
addresses are: " & Format$(Centronic(1), "###") & " " & Format$(Centronic(2), "###") & " " &
Format$(Centronic(3), "###") & " " & Format$(Centronic(4), "###") & "Decimal", 48, "Centronic ports (LPT) on
your PC")
    lpt_number = Val(InputBox$("Input 1, 2, 3 or 4 to select a Centronic port (LPT) for the Centronic Experimental
board", "Select LPT ports"))
    P_address = Centronic(lpt_number)
    For i = 0 To 11
        status(i) = 0
    Next i
    dummy = Write_control(P_address, 137) 'Configure the 8255, port A and B as outputs, port C as input
    Label2(0).Caption = "No of installed LPTs: " & Format(lpt_number)
    Label4(0).Caption = "Address of LPT: " & Format(P_address)
End Sub

Private Sub Frame1_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    Label3.Caption = "Select output status of Port A"
End Sub

Private Sub Frame2_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    Label3.Caption = "Select output status of Port B"
End Sub

Private Sub Frame3_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    Label3.Caption = "View status of Port C"
End Sub

```

4.4 Using other programming languages

If other programming languages such as BASIC or C++ are used, users have to write their own I/O drivers. I/O functions written in TP6 are listed in the Program List 1. They are useful examples.

All I/O functions involve input or output data through the Centronic port. A brief description of a Centronic port is given below:

A Centronic port consists of three separate ports. They are Data port (8 bits), Control port (5 bits) and Status port (4 bits). Data and Control ports are output ports and Status port is an input port. If the base I/O address for the Data port is *P_addr*, the corresponding I/O Control port and the Status

port are: P_addr+2 and P_addr+1 . Commands controlling output and input through these three port are described as follows:

Tasks	Turbo Pascal 6	Qbasic	C++
Output from Data port	PORT[P_addr]=Data;	OUT P_addr, Data	OutP P_addr,Data
Output from Control port	PORT[P_addr+2]=Data	OUT P+addr_2,Data	OutP P_addr+2,Data
Input from Status port	Data:=PORT[P_addr+1]	Data=INP(P_addr+1)	Data=INP(P_ddr+1)

5. DOS program - CIODos

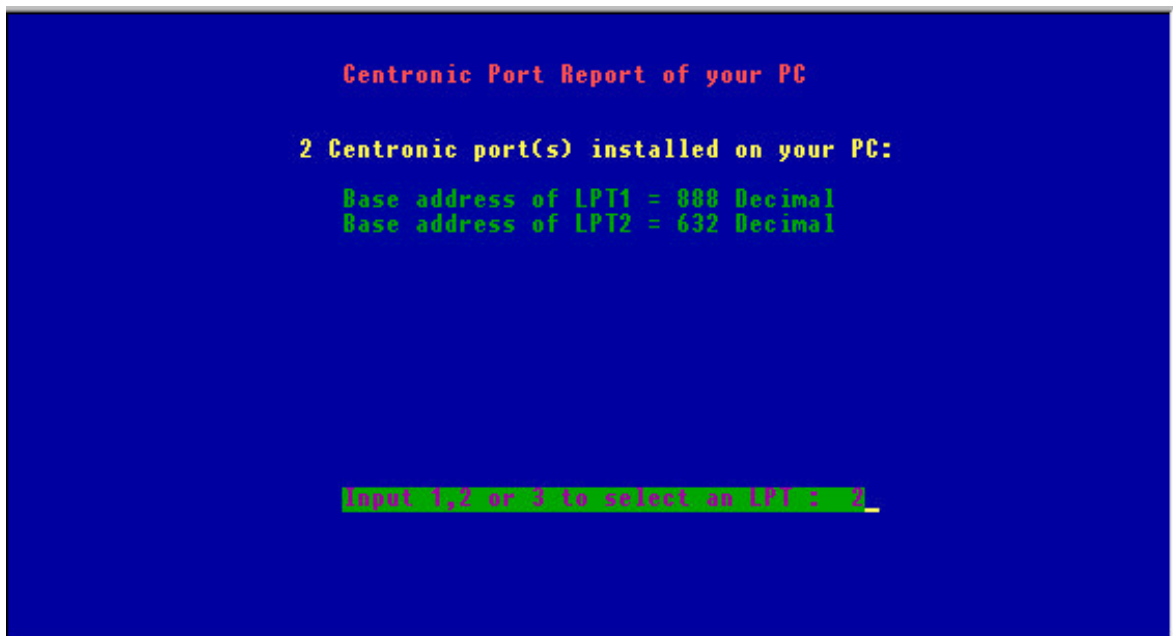
“CIODos” is a DOS-based user software. It allows users to test the functionality of the CIO-1 card.

3.1 Start the program

You start the “CIODos” program in the usual way. In DOS environment, type in “CIODos” in the directory containing the file and press Return. In Window 95 environment, the program can be started from START/RUN or from EXPLORER.

3.2 Select a Centronic port

The first screen reports the number of Centronic ports installed on your computer and allows you to select a port to be used with the CIO-1 card. Users key in “1”, “2” or “3” followed by a RETURN to select LPT1, LPT2 or LPT3 (Figure 2). Press RETURN to accept. Next, the program checks the data transfer speed of the card.



```
Centronic Port Report of your PC

2 Centronic port(s) installed on your PC:

Base address of LPT1 = 880 Decimal
Base address of LPT2 = 632 Decimal

Input 1,2 or 3 to select an LPT : 2_
```

Figure 2 The window reports number of LPT ports installed and allows you to select one for the card

3.3 Main screen

After you have selected the LPT port and after a speed check, a window shown in Figure 3 appears. The I/O lines are configured as follows:

- Port A as input (8-bit)
- Port B and C as outputs (8-bits)
- The control word to achieve the above configuration is 144 decimal

This window allows users to test each bit of the output ports. Input "1", "2" or "8" to select a bit to be tested. The selected bit in Ports B and C produces a square wave (about 1 Hz) continuously. An LED, a logic probe or a multimeter can be used to test the outputs. The data present at Port A is read into the computer and is shown on the window. To test another bit, just press a key and a new bit can be keyed in. Input "9" to quit the program.

```
Centronic port - 8255 card testing program

Port B and C are all configured as Outputs
Port A is configured as Input

The control word sent to the 8255 PPI is 144 decimal
Input the bit number to be tested (1 to 8, 9 to quit) : 1

Data from Port A =146_

The selected bit of Port B and C produces a square wave
Press Return to continue
```

Figure 3 Main screen for testing outputs (Ports B and C) and input (Port A)

6. Windows program - CIOWin

"CIOWin" is a Windows95/98 application software. It allows users to check the functionality of the CIO card in Windows environment.

16-bit and 32-bit CioWin are installed from the supplied disks (or CD). Following the standard installation procedures to install these two program packages.

6.1 Start the program

In Windows 95/98 environment, CioWin program is started by clicking **Start** and is followed by selecting **Run** in the Start menu. Then type in the program name with the full path (for example C:\CIO\CioWin). CioWin can be also started from the Windows Explorer.

6.2 Run the program

After the program starts, the window shown in Figure 4 appears on the screen. It reports the number of LPT ports installed on your PC. Click OK to continue. Window shown in Figure 5 asks users to select an LPT port.

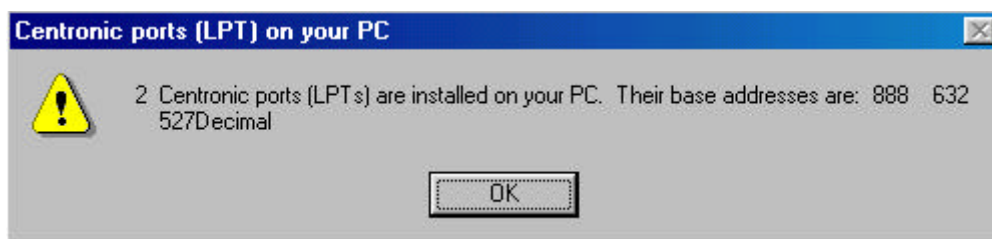


Figure 4 Window reporting the number of LPT ports installed on you PC (16-bit version)

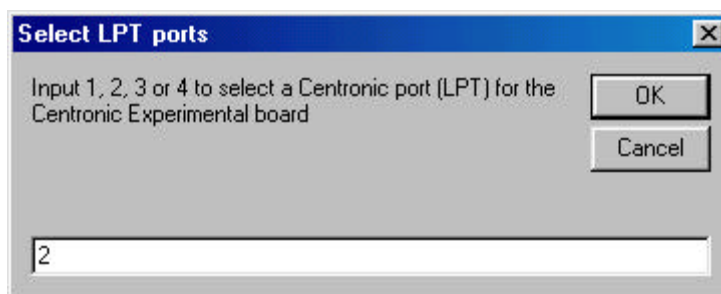


Figure 5 Window allowing users to select an LPT port (16-bit version)

For the 32-bit version, the CIOWin will not report the number of the LPT ports installed. It will ask uses to input the base address of the LPT port to be used. The base address for all the installed LPT ports on your PC can be found in the Hardware/System/Control Panel. In most cases, the the base addresss for LPT1 and LPT2 are 888 decimal and 956 decimal. This may vary for PC to PC.

Figure 6 is the window allows users to output data to Ports A and B and read data from Port C. Click "0/1" buttons to change the output status. Press "Output and Input Button" to output the data to Ports A and B. Digital data present at Port C is read into the pc and status shown on the screen. 16-bit version and 32-bit version have the same screen.

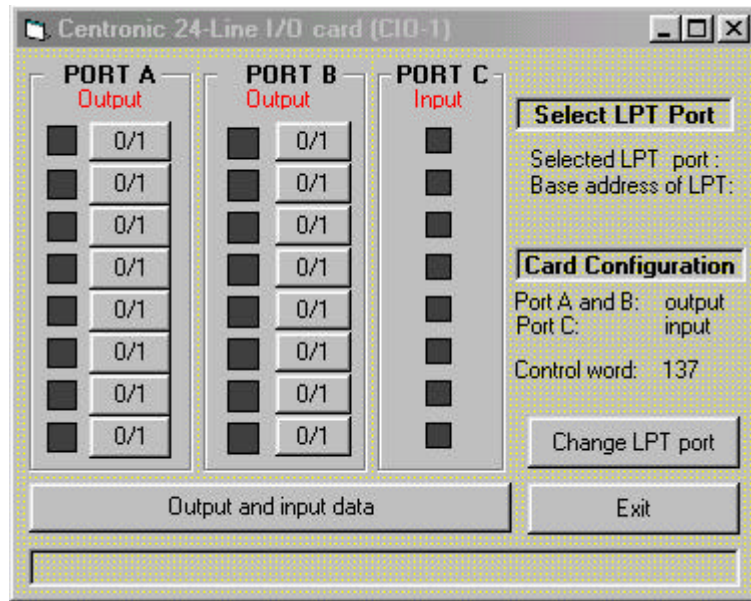


Figure 6 Windows to output and input data

7. CIOWin for NT/2000/XP

"CIOWin" is written for Windows95 98 and Me.

Problem will rise if CIOWin runs in NT/2000 and Windows XP. This is because that NT/2000/XP is very strict in controlling over I/O ports. Unlike Windows 95/98/Me, Windows NT/2000/XP will cause an exception (Privileged Instruction) if an attempt is made to access a port that you are not privileged to talk too.

The I/O permission bitmap can be used to allow programs not privileged enough the ability to access the I/O ports. When an I/O instruction is executed, the processor will first check if the task is privileged enough to access the ports. Should this be the case, the I/O instruction will be executed. However if the task is not allowed to do I/O, the processor will then check the I/O permission bitmap.

The I/O permission bitmap, as the name suggests uses a single bit to represent each I/O address. If the bit corresponding to a port is set, then the instruction will generate an exception however if the bit is clear then the I/O operation will proceed. This gives a means to allow certain processes to access certain ports. There is one I/O permission bitmap per task.

6.1 Install Porttalk Driver

Copy porttalk.sys to winnt/system32/drivers or (Windows/system32/drivers on XP) and then double click porttalk.reg to enter the required keys into your registry. This will load the driver automatically at next reboot.

6.2 Allowio

Usage : AllowIO <Hex Adresse(s)> <executable.exe> <Switches>

Example: Using Allowio to grant access to IO ports 0x42, 0x43 and 0x61
 C:\porttalk\AllowIO>allowio 0x42 0x43 0x61 YourProg.exe

If you don't know what addresses a certain program uses, you can grant access to all ports.

Example: C:\porttalk>allowio CIOWin.exe /a